WHAT IS CLAIMED IS:

1. A cellular engine for a data processing system, said engine comprising:

a data device having *n*- cells, each of said *n*- cells being able to store *m* bits;

a vector memory containing *p*- vectors, each of said *p*- vectors having a storage capacity of *n* x *m*-bits;

a control interconnection network that classifies each of said *n*- cells in dependence upon a local state of each of said *n*- cells;

an instruction register for accepting an instruction issued from a controller;

a clock device for outputting a synchronizing clock signal comprised of a predetermined number of clock cycles per second, said clock device outputting said synchronizing clock signal to said data device and said vector memory; and

wherein said engine globally communicates said instruction to all of said *n*-cells simultaneously within one of said clock cycles, said instruction being executed in parallel by selected cells within said data device, all within one of said clock cycles, in accordance with said classification of each of said *n*- cells by said control interconnection network.

2. The cellular engine for a data processing system according to claim 1, wherein:

said instruction is executed in parallel by all of said *n*- cells within said data device.

3. The cellular engine for a data processing system according to claim 1, further comprising:

a data interconnection network that connects each cell to its right and left neighbors respectively.

4. The cellular engine for a data processing system according to claim 1, wherein:

said control interconnection network classifies each of said *n*- cells in dependence upon both a local state of each of said *n*- cells and a global state of all of said *n*- cells.

5. The cellular engine for a data processing system according to claim 4, wherein:

each of said *n*- cells include a state field and a data field, said state field comprising a marker bit for encoding a local state of each of said *n*- cells; and

wherein said maker bit is in one of a marked state and a non-marked state.

6. The cellular engine for a data processing system according to claim 5, wherein:

said state field is modified using associative mechanisms as implemented by said execution of said instruction belonging to a specific subset of instructions

7. The cellular engine for a data processing system according to claim 6, wherein:

said data field is modified by executing logic and arithmetic instructions in said *n*- cells in accordance with said classification of each of said *n*- cells by said control interconnection network.

8. The cellular engine for a data processing system according to claim 3, wherein:

both of said data interconnection network and said control interconnection network are expandable.

9. The cellular engine for a data processing system according to claim 1, wherein:

said data device is an associative memory device.

10. The cellular engine for a data processing system according to claim 9, wherein:

each of said *n*- cells in said associative memory device include a processing circuit.

11. An engine for a data processing system, said engine comprising:

a memory device containing *n*-cells;

a controller for selectively issuing an instruction to said memory device;

a cell classification device which operates in association with a local state of each of said *n*-cells;

a clock device for outputting a synchronizing clock signal comprised of a predetermined number of clock cycles per second, said clock device outputting said synchronizing clock signal to said memory device and said controller; and

wherein said engine globally communicates said instruction to all *n*-cells simultaneously, within one of said clock cycles; and

wherein said instruction is executed by selected cells within said memory device in dependence upon said local state of said *n*- cells as directed by said cell classification device, said execution of said instruction occurring simultaneously in each of said selected cells within one of said clock cycles.

12. The engine for a data processing system according to claim 11, wherein:

said instruction is executed by all of said *n*- cells within said memory device.

13. The engine for a data processing system according to claim 11, wherein:

said memory device is not conventionally addressed.

14. The engine for a data processing system according to claim 11, wherein:

said memory device is an associative memory device.

15. The engine for a data processing system according to claim 11, wherein:

each of said *n*- cells includes a processing circuit.

16. The engine for a data processing system according to claim 15, wherein:
said processing circuit has an accumulator.

17. The engine for a data processing system according to claim 11, wherein:
each of said *n*-cells includes a field having a marker bit; and
said local state reflects one of a marked state and a non-marked state of said marker bit.

18. The engine for a data processing system according to claim 11, wherein:
said cell classification device operates in association with a global state of said *n*-cells; and
wherein said instruction is executed by selected cells within said memory device in dependence upon said local state and said global state of said *n*- cells as directed by said cell classification device, said execution of said instruction occurring simultaneously in each of said selected cells within one of said clock cycles.

19. The engine for a data processing system according to claim 18, wherein:
said global state utilized by said classification device is determined by said local states of all said *n* -cells.

20. The engine for a data processing system according to claim 17, wherein:
said instruction is executed only within those *n*-cells having said marker bit set to said marked state.

21. The engine for a data processing system according to claim 11, wherein:
each of said *n*-cells includes a field having a plurality of marker bits.

22. A data processing system, said data processing system comprising:

an associative memory device containing *n*-cells, each of said *n*-cells including a processing circuit and *m*- bits of memory capacity;

a controller for issuing one of a plurality of instructions to said associative memory device;

a clock device for outputting a synchronizing clock signal comprised of a predetermined number of clock cycles per second, said clock device outputting said synchronizing clock signal to said associative memory device and said controller; and

wherein said controller globally communicates one of said plurality of instructions to all of said *n*-cells simultaneously, within one of said clock cycles.

23. The data processing system of claim 22, further comprising:

a classification device for selectively operating in association with a local state of each of said *n*-cells; and

wherein one of said plurality of instructions is executed by selected cells within said associative memory device in dependence upon said local state of said *n*- cells as directed by said classification device, said execution of said instruction occurring simultaneously in each of said selected cells within one of said clock cycles.

24. The data processing system of claim 23, wherein:

each of said *n*-cells include a state field and a data field, said state field comprising a marker bit for encoding a local state of each of said *n*- cells, thereby indicating one of a marked state and a non-marked state of each of said *n*-cells.

25. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'load line immediate' command whereby the contents of all of said *n*- cells in said marked state are replaced with data indicated by said 'load line immediate' command.

26. The data processing system of claim 25, further comprising:

a vector memory containing *p*- vectors, each of said *p*- vectors containing *n*-elements having *m*-bits each; and

said data indicated by said 'load line immediate' command corresponds to one of said *p*- vectors.

27. The data processing system of claim 24, further comprising:

one of said plurality of instructions issued by said controller is a 'store line immediate' command whereby the contents of all of said *n*- cells in said marked state are saved to a memory vector indicated by said 'store line immediate' command.

28. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'left limit' command whereby a left limit of a search space is set to a leftmost cell of said *n*- cells in said marked state.

29. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'right limit' command whereby a right limit of a search space is set to a leftmost cell of said *n*- cells in said marked state.

30. The data processing system of claim 24, wherein:

said controller may dynamically limit a search space within said *n*- cells.

31. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'find' command whereby each of said *n*- cells holding values equal to an argument indicated by said 'find' command is identified; and

wherein said 'find' command sets said marker bit to said marked state in each of said *n*- cells located to the right of said identified *n*- cells, and sets said marker bit to said non-marked state in all other of said *n*- cells.

32. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'match' command whereby each of said *n*- cells having a marker bit in said marked state and having said data field matching an argument indicated by said 'match' command, is identified; and

wherein said 'match' command sets said marker bit to said marked state in each of said *n*- cells following said identified *n*- cells, and sets said marker bit to said non-marked state in all other of said *n*- cells.

33. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'find and mark left' command whereby each of said *n*- cells holding values equal to an argument indicated by said 'find and left mark' command is identified; and

wherein said 'find' command sets said marker bit to said marked state in each of said *n*- cells located to the left of said identified *n*- cells, and sets said marker bit to said non-marked state in all other of said *n*- cells.

34. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'match and mark left' command whereby each of said *n*- cells having a marker bit in said marked state and having said data field matching an argument indicated by said 'match' command, is identified; and

wherein said 'match' command sets said marker bit to said marked state in each of said *n*- cells located to the left of said identified *n*- cells.

35. The data processing system of claim 28, wherein:

one of said plurality of instructions issued by said controller is a 'markall' command whereby said marker bit of each of said *n*- cells within said search space is set to said marked state.

36. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is an 'addmark' command whereby said marker bit of each of said *n*- cells containing a value equal to an argument indicated by said 'addmark' command is set to said marked state; and

said 'addmark' command does not affect said marker bit of any other of said *n*- cells.

37. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'mark' command whereby said marker bit of each of said *n*- cells containing a value equal to an argument indicated by said 'mark' command is set to said marked state; and

said 'mark' command sets said marker bit to said non-marked state in each of said *n*- cells not containing a value equal to an argument indicated by said 'mark' command.

38. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'clr' command whereby said marker bit of each of said *n*- cells containing a value equal to an argument indicated by said 'clr' command is set to said non-marked state.

39. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'clear first' command whereby said leftmost of said *n*- cells having said marker bit set to said marked state is set to said non-marked state.

40. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'trace' command whereby said marker bit of all of said *n*- cells is duplicated leftward in said associative memory device.

41. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'keepl' command whereby said marker bit of all of said *n*- cells except for a rightmost of said *n*- cells in said marked state is set to said non-marked state.

42. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'clrl' command whereby said marker bit of a rightmost of said *n*-cells in said marked state is set to said non-marked state.

43. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'left' command whereby all of said marker bits for said *n*- cells are shifted leftward by one.

44. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'right' command whereby all of said marker bits for said *n*- cells are shifted rightward by one.

45. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'cright' command whereby all of said marker bits for said *n*- cells that are in said marked state are shifted rightward by one to a next-right cell unless said next-right cell contains a value equal to an argument indicated by said 'cright' command; and

said 'cright' command replaces said data field of said next-right cell with a predetermined value when said next-right cell contains a value equal to an argument indicated by said 'cright' command.

46. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'cleft' command whereby all of said marker bits for said *n*- cells that are in said marked state are shifted leftward by one to a next-left cell unless said next-leftt cell contains a value equal to an argument indicated by said 'cleft' command; and

said 'cleft' command replaces said data field of said next-left cell with a predetermined value when said next-left cell contains a value equal to an argument indicated by said 'cleft' command.

47. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'nop' command whereby no operation is executed.

48. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'reset' command whereby said data fields of said *n*- cells are assigned a value indicated by said 'reset' command.

49. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'get' command whereby a value stored in a leftmost cell of said *n*- cells in said marked state is outputted from said associative memory device; and

said marker bit of said leftmost cell in said marked state is shifted to the right by one cell.

50. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'back' command whereby a value stored in a leftmost marked cell is outputted from said associative memory device; and

said marker bit of said leftmost marked cell is shifted to the left by one cell.

51. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'set' command whereby a value indicated by said 'set' command is stored in a leftmost of said *n*- cells in said marked state.

52. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'setall' command whereby a value indicated by said 'setall' command is stored in all of said *n*- cells in said marked state.

53. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is an 'ins' command whereby a value indicated by said 'ins' command is stored in one of said *n*- cells at an insertion point located prior to a leftmost of said *n*- cells in said marked state; and

said 'ins' command rightwardly shifts a content of each of said n- cells located to the right of said insertion point by one.

54. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'del' command whereby a data field of a leftmost of said *n*- cells in said marked state is deleted; and

said 'del' command leftwardly shifts a content of each of said *n*- cells located to the right of said leftmost marked cell by one.

55. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'cpr' command whereby contents of all of said *n*- cells in said marked state are rightwardly copied by one.

56. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'cpl' command whereby contents of all of said *n*- cells in said marked state are leftwardly copied by one.

57. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'ccpr' command whereby contents of all of said *n*- cells in said marked state are rightwardly copied by one, including copying of said marked state; and

said 'ccpr' command does not copy said marked state of said *n*- cells whose said contents equals a value indicated by said 'ccpr' command.

58. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'ccpl' command whereby contents of all of said *n*- cells in said marked state are leftwardly copied by one, including copying of said marked state; and

said 'ccpr' command does not copy said marked state of said *n*- cells whose said contents equals a value indicated by said 'ccpl' command.

59. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'ld' command whereby a value indicated by said 'ld' command is loaded into said data field in all of said *n*- cells in said marked state.

60. The data processing system of claim 24, further comprising:

a vector memory containing *p*- vectors; and

one of said plurality of instructions issued by said controller is a 'st' command whereby contents of said data field in all of said *n*- cells in said marked state are copied to a corresponding location in one of said *p*- vectors as indicated by said 'st' command.

61. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is an 'add' command whereby a value indicated by said 'add' command is added to said data field in all of said *n*- cells in said marked state.

62. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'fadd' command whereby a value indicated by said 'fadd' command is added to said data field in all of said *n*- cells in said marked state; and

wherein a leftmost bit of said data field of a cell within said associative memory device positioned to the right of each of said *n*- cells in said marked state acts as a carry bit.

63. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'sub' command whereby a value indicated by said 'sub' command is subtracted from said data field in all of said *n*- cells in said marked state.

64. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'fsub' command whereby a value indicated by said 'fsub' command is subtracted from said data field in all of said *n*- cells in said marked state; and

wherein a leftmost bit of said data field of a cell within said associative memory device positioned to the right of each of said *n*- cells in said marked state acts as a carry bit.

65. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'half' command whereby said data field in all of said *n*- cells in said marked state is divided by two.

66. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is an 'fhalf' command whereby said data field in all of said *n*- cells in said marked state is divided by two; and

wherein 1 is added to a leftmost bit of said data field if a least significant data bit of a cell within said associative memory device positioned to the left of each of said *n*- cells is in said marked state.

67. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'lt' command whereby a value indicated by said 'lt' command is compared to said data field in all of said *n*- cells in said marked state;

said 'lt' command sets said marker bit of said *n*- cells in said marked state to said non-marked state if said value indicated by said 'lt' command is greater than said data field; and

said 'lt' command sets to 1 a leftmost bit of said data field of said *n*- cells in said marked state if said value indicated by said 'lt' command is less than said data field.

68. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'flt' command whereby a value indicated by said 'flt' command is compared to said data field in all of said *n*- cells in said marked state;

said 'flt' command sets said marker bit of said *n*- cells in said marked state to said non-marked state if said value indicated by said 'flt' command is greater than said data field and a leftmost bit of said data field of a cell within said associative memory device positioned to the left of each of said *n*- cells in said marked state is zero;

said 'flt' command sets to 1 a leftmost bit of said data field of said *n*- cells in said marked state if said value indicated by said 'flt' command is less than said data field;

said 'flt' command sets to 1 a leftmost bit of said data field of said *n*- cells in said marked state if said leftmost bit of a cell within said associative memory device positioned to the left of each of said *n*- cells in said marked state is 1; and

said 'flt' command sets to 0 said leftmost bit of a cell within said associative memory device positioned to the left of each of said *n*- cells in said marked state.

69. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'gt' command whereby a value indicated by said 'gt' command is compared to said data field in all of said *n*- cells in said marked state;

said 'gt' command sets said marker bit of said *n*- cells in said marked state to said non-marked state if said value indicated by said 'gt' command is less than said data field; and

said 'gt' command sets to 1 a leftmost bit of said data field of said *n*- cells in said marked state if said value indicated by said 'gt' command is greater than said data field.

70. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'fgt' command whereby a value indicated by said 'fgt' command is compared to said data field in all of said *n*- cells in said marked state;

said 'fgt' command sets said marker bit of said *n*- cells in said marked state to said non-marked state if said value indicated by said 'fgt' command is less than said data field and a leftmost bit of said data field of a cell within said associative memory device positioned to the left of each of said *n*- cells in said marked state is zero;

said 'fgt' command sets to 1 a leftmost bit of said data field of said *n*- cells in said marked state if said value indicated by said 'fgt' command is greater than said data field;

said 'fgt' command sets to 1 a leftmost bit of said data field of said *n*- cells in said marked state if said leftmost bit of a cell within said associative memory device positioned to the left of each of said *n*- cells in said marked state is 1; and

said 'fgt' command sets to 0 said leftmost bit of a cell within said associative memory device positioned to the left of each of said *n*- cells in said marked state.

71. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'test' command whereby for all of said *n*- cells in said marked state that contain a value indicated by said 'test' command, said 'test' command sets said marker bit to said non-marked state; and

said 'test' command assigns a predetermined value to said data field of all of said *n*- cells having said indicated value if a leftmost bit of a cell to the left of all of said *n*- cells having said indicated value is 1; and

said 'test' command clears a leftmost bit of said cell to the left of all of said *n*- cells having said indicated value, if said leftmost bit is 1.

72. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is an 'and' command whereby for all of said *n*- cells in said marked state, said 'and' command accomplishes a bitwise AND of a value indicated by said 'and' command and said data field in all of said *n*- cells in said marked state.

73. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is an 'or' command whereby for all of said *n*- cells in said marked state, said 'or' command accomplishes a bitwise OR of a value indicated by said 'or' command and said data field in all of said *n*- cells in said marked state.

74. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'xor' command whereby for all of said *n*- cells in said marked state, said 'xor' command accomplishes a bitwise XOR of a value indicated by said 'xor' command and said data field in all of said *n*- cells in said marked state.

75. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'cond' command whereby for all of said *n*- cells in said marked state, said 'cond' command switches their marked state to said non-marked state if a result of a bitwise AND operation between a value indicated by said 'cond' command and said data field of all of said *n*- cells in said marked state is 0.

76. The data processing system of claim 24, wherein:

one of said plurality of instructions issued by said controller is a 'ncond' command whereby for all of said *n*- cells in said marked state, said 'ncond' command switches their marked state to said non-marked state if a result of a bitwise AND operation between a value indicated by said 'ncond' command and said data field of all of said *n*- cells in said marked state is not 0.

77. The data processing system of claim 71, wherein:

one of said plurality of instructions issued by said controller is an 'index' command whereby for all of said *n*- cells in said marked state, said 'index' command assigns a value to each of said marked *n*- cells indicative of each of said marked *n*- cells position relative to a leftmost of said *n*- cells.

78. The data processing system of claim 22, wherein:

said classification device determines a global state of each of said *n*-cells; and

wherein said one of said plurality of instructions is executed by selected cells within said associative memory device in dependence upon both said local state and said global state of said *n*- cells, said execution of said instruction occurring simultaneously in each of said selected cells within one of said clock cycles.

79. The data processing system of claim 22, wherein:

said associative memory device allows the use of variable-length key fields.

80. The data processing system of claim 78, wherein:

said variable-length key fields include a data field and a key field, each of said *n*- cells having said data field and said key field; and

wherein data stored in each of said *n*- cells is alternatively considered as part of said data field and said key field at different times during execution of one of said plurality of instructions.

81. A method of processing data, said method comprising the steps of:

forming an associative memory device to contain *n*-cells;

configuring each of said *n*-cells to include a processing circuit;

issuing one of a plurality of instructions from a controller to said associative memory device;

utilizing a clock device for outputting a synchronizing clock signal comprised of a predetermined number of clock cycles per second, said clock device outputting said synchronizing clock signal to said associative memory device and said controller; and

globally communicating one of said plurality of instructions from said controller to all of said *n*-cells simultaneously, within one of said clock cycles.

82. A data processing system, said data processing system comprising:

a memory device containing *n*-cells, each of said *n*-cells including a processing circuit;

a controller for issuing one of a plurality of instructions to said associative memory device;

a clock device for outputting a synchronizing clock signal comprised of a predetermined number of clock cycles per second, said clock device outputting said synchronizing clock signal to said associative memory device and said controller; and

wherein said controller globally communicates one of said plurality of instructions to all of said *n*-cells simultaneously, within one of said clock cycles.